

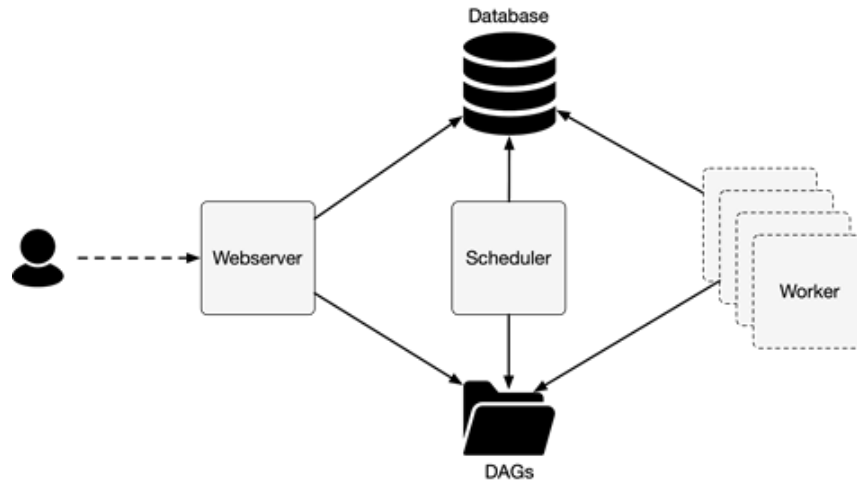


Orquestração de Fluxos de Dados com Apache Airflow

Arquitetura do Apache Airflow

O **Apache Airflow** é uma plataforma de código aberto usada para programar e monitorar fluxos de trabalho. Ele permite que os usuários orquestram pipelines de dados complexos, definindo tarefas e suas dependências.

Descrevemos abaixo uma visão geral da arquitetura do Apache Airflow:



DAG (Directed Acyclic Graph): No coração do Airflow está o conceito de DAG, que é uma coleção de todas as tarefas que você deseja executar, organizadas de forma que reflitam suas relações e dependências.

Scheduler: O agendador monitora todos os DAGs e garante que o que deve ser executado seja executado. Ele verifica as dependências e inicia a execução de uma tarefa se todas as suas dependências tiverem sido atendidas.

Work Queue: Para a execução de tarefas, o agendador coloca as tarefas em uma fila de trabalho. Os trabalhos em uma fila são processados por workers.

Worker Process: Os workers pegam as tarefas dessa fila de trabalho e as executam. Uma vez que uma tarefa é executada, o estado da tarefa é registrado no backend de metadados.

Metastore Database: É um banco de dados que o Airflow usa para manter o estado e os detalhes de todos os DAGs, tarefas e suas execuções. O banco de dados de metadados é essencial para a resiliência e a capacidade de recuperação do Airflow.

Web Interface: O Airflow vem com uma interface web rica que permite aos usuários visualizar os DAGs, verificar o status das execuções, visualizar logs, forçar a execução de tarefas e muito mais.



Orquestração de Fluxos de Dados com Apache Airflow

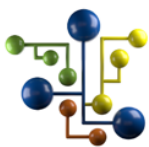
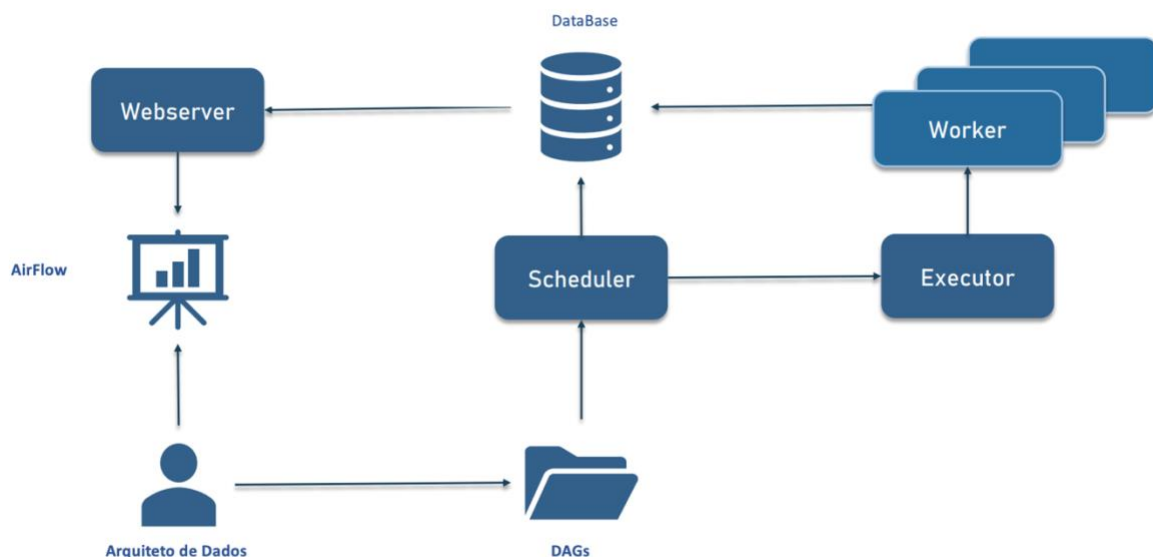
Executor: O executor determina como as tarefas são executadas em workers. Existem diferentes tipos de executores disponíveis no Airflow como SequentialExecutor, LocalExecutor, CeleryExecutor e KubernetesExecutor.

Plugins: O Airflow é extensível e permite que os usuários escrevam plugins personalizados para operadores, sensores, hooks, interfaces e muito mais.

Hooks & Operators: Os hooks são interfaces para sistemas externos, enquanto os operadores são blocos de construção que realizam uma ação. Por exemplo, um PythonOperator executaria um script Python, enquanto um PostgresOperator executaria uma consulta SQL em um banco de dados PostgreSQL.

XComs: É um mecanismo que permite que as tarefas troquem mensagens ou pequenas quantidades de dados. É útil quando uma tarefa precisa passar algum tipo de informação para outra tarefa.

A arquitetura do Airflow é projetada para ser extensível e escalável, permitindo que os usuários orquestram fluxos de trabalho complexos em sistemas distribuídos.



Equipe DSA

Muito Obrigado!
Continue Trilhando Uma Excelente Jornada de Aprendizagem.